Amendments to the Specification:

Please delete the paragraph beginning on page 1, line 6 and replace with the following:

The present invention relates generally to Java^{TM} virtual machines ("Java" is a trademark of Sun Microsystems, Inc.), and more particularly to analyzing thread dumps in a Java virtual machine.

Please delete the paragraph beginning on page 1, line 11 and replace with the following:

Ever since the rise of the Internet, the use of the Java Platform, from Sun Microsystems, Inc., by the software development community has increased significantly. It is partly due to the advantages of its own Java Programming Language, which is object-oriented, distributed, multithreaded and portable. An Object-oriented programming language such as Java programming language provides programmers flexibility to create modules that do not need to be changed when a new type of object is added; this enables an object to inherit the characteristics of another object, which contributes to reusable software. As a result, development time can be shortened and developers can spend more time on other aspects of the software life cycle.

Please delete the paragraph beginning on page 3, line 5 and replace with the following:

As discussed above, another important feature of the Java Platform is its support for multithreading at the language level. **Figure 3C** depicts multithreading in a block diagram. A thread is a basic unit of program execution. At any given time, a program can have several threads running concurrently, each thread performing a different job. The Java language accomplishes such tasks by synchronization, which coordinates activities and data access among multiple threads. The Java Virtual Machine uses a mechanism named Monitors to support synchronization. There is a lock associated with every object or resource in a Java environment. If multiple threads want to operate on the same object, a monitor is used to provide a way for the threads to independently work on the object without interfere with each other. When a thread wants to acquire a shared object, a code segment within a program identified with the synchronized keyword is used to associate a lock with every object that has synchronized code. Once the lock for the object is obtained by performing a lock operation in the JVM, the body of the code segment is then executed. The thread becomes the owner of the object **314**, hence an active thread **318**. If during the time of execution of the active thread another thread wants to claim ownership of the monitor, it must wait in the entry set **312** of the object along with the other threads already waiting. Once the active thread is finished executing the critical region, it can release the monitor in two ways: it can complete the execution or it can issue a wait command. In the prior case, the thread can simply exit the monitor **320**. Alternatively, by issuing the wait command, the active thread becomes a waiting thread in the wait set **316**. If the former owner did not issue a notify command before it releases the monitor, only threads in the entry set will compete to acquire the monitor. If the former owner did execute a notify, then the entry set along with any threads in the wait set will compete to acquire the monitor. If a thread in the wait state wins, it then exits the wait set and reacquires the monitor. Once again, it becomes an active thread.

Please delete the paragraph beginning on page 10, line 27 and replace with the following:

An operating system runs on processor **202** and is used to coordinate and provide control of various components within data processing system **200** in **Figure 2**. The operating system may be a commercially available operating system such as Windows 2000, which is available from Microsoft Corporation. An object oriented programming system such as Java <u>programming language</u> may run in conjunction with the operating system and provides calls to the operating system from Java programs or applications executing on data processing system **200**. "Java" is a trademark of Sun Microsystems, Inc. Instructions for the operating system, the object-oriented programming system, and applications or programs are located on storage devices, such as hard disk drive **226**, and may be loaded into main memory **204** for execution by processor **202.**